



Community

Welcome, Guest [Login](#) [Register](#)[Home](#) > [Community](#) > [LabVIEW Development Best Practices](#) > [Documents](#)[Up to Documents in LabVIEW Development Best Practices](#)

Source Code Control: Using TortoiseSVN (Subversion) with LabVIEW for Diff and Merge Operations

VERSION 9

Created on: Dec 10, 2008 1:48 AM by ElijahK - Last Modified: Feb 15, 2012 11:35 AM by ElijahK

Note: for more general information on group development and configuration management best practices and the use of other source code control tools with LabVIEW, please consult the following white paper: [Software Configuration Management in LabVIEW](#).

Overview

Most source code control providers have built-in diff and merge capabilities for text-based files, but they also typically allow users to override the default utilities for specific file-types. The binary nature of LabVIEW file-types requires the use of custom utilities for diff and merge, which are available in the professional version of LabVIEW 2009 or later.

Calling graphical merge from an external application requires the ability to pass parameters to the command line. In windows, the executable to be run for [Graphical Diff](#) is available at the following location: **C:\Program Files\National Instruments\Shared\LabVIEW Compare\LVCompare.exe**. The analogous executable for [Graphical Merge](#) is available at **C:\Program Files\National Instruments\Shared\LabVIEW Merge\LVMerge.exe**.

Both executables offer a variety of command-line parameters that can be used to customize the behavior and results of the utility. As an example, these settings can be used to exclude changes that are purely cosmetic and do not affect the behavior of code.

Configuring TortoiseSVN for Graphical Diff

1. Right-click in Windows Explorer to expose the TortoiseSVN options
2. Mouse-over 'TortoiseSVN >> Settings'
3. In the list of settings on the left, select 'Diff Viewer' under 'External Programs'
4. On the right side of the dialog, select 'Advanced'
5. When the 'Advanced diff settings' dialog appears, select 'Add'
6. In the dialog that appears, type .vi as the extension
7. Where it prompts you for the path to the external program, type the following: **"C:\Program Files\National Instruments\Shared\LabVIEW Compare\LVCompare.exe" %mine %base -nobdcosm -nobdpos**
8. Repeat this operation for the file-type .ctl

Configuring TortoiseSVN for Graphical Merge

1. Right-click in Windows Explorer to expose the TortoiseSVN options
2. Mouse-over 'TortoiseSVN >> Settings'
3. In the list of settings on the left, select 'Merge Tool' under 'External Programs'
4. On the right side of the dialog, select 'Advanced'
5. When the 'Advanced merge settings' dialog appears, select 'Add'
6. In the dialog that appears, type .vi as the extension
7. Where it prompts you for the path to the external program, type the following: **"C:\Program Files\National Instruments\Shared\LabVIEW Merge\LVMerge.exe" %base %mine %theirs %merged**
8. Repeat this operation for the file-type .ctl

Example

To ensure that TortoiseSVN is downloaded and installed correctly, download the [software engineering technical manual](#) and follow the steps in Exercise 1.

11298 Views

Tags: [control](#), [configuration](#), [labview](#), [management](#), [source](#), [code](#), [software](#), [scc](#), [tutorial](#)

Like (0)

Actions

[View print preview](#)[Login or Create Profile](#)

Bookmarked By (4)

View: 

More Like This

[Source Code Control and Team-Based Development Demo Material](#)[Subversion and LabVIEW: Tips, Tricks, and Pitfalls](#)[8 LabVIEW Large LabVIEW Application Development](#)[Software Engineering Technical Manual and Exercises for LabVIEW](#)[Hands-On: Software Engineering With LabVIEW](#)

Incoming Links

[Source Code Control and Group Development Resources for LabVIEW](#)[May 2011 Blog of the Month: LabVIEW for Large Application Development](#)[How To Make Your Online Group a Success](#)[7/21 Meeting Source Code Control Presentation](#)

More by ElijahK

[Measurement Abstraction and Model-View-Controller \(MVC\) Project with Actor Framework in LabVIEW](#)[Using Variant Attributes for High-Performance Lookup Tables in LabVIEW](#)[Thoughts on Software Sessions at NIWeek](#)[Software Engineering Technical Manual and Exercises for](#)

Average User Rating

(1 rating)

Comments (12)





Lul May 9, 2011 8:20 AM

Elijah,

thx for this summary. I've got a Q:
 Since most of my workhorse projects are still under LV711, I am not that familiar with using the project explorer and I simply use TortoiseSVN to externally doing my revision control. And here's my Q coming from.
 I have a lot of apps that share several 'libraries', mostly from user.lib and instr.lib. Doing revision control I have to control those libs as well and shall relate different versions of those libs to the different revisions of the apps. I could copy the linked modules from the libs into the respective development working copy - and loose the links. Or I can hardlink into the lib, but this would not provide specific revision info. Or I could use SVN:Externals, as Jim Kring mentions at <http://thinking.com/2008/10/21/set-the-revision-of-your-svnexternals-or-else/>
 What would you suggest? Do you have a better solution?

TIA!

Greetings from Germany!

--

Uwe

Like (0)

[LabVIEW](#)

[Unit Test Framework Toolkit](#)
[Users: We want your feedback!](#)

[View ElijahK's profile](#)



FabiolaDelaCueva May 9, 2011 8:56 AM (in response to Lul)

Lul,

I haven't worked with LV7 in a while, but here are some ideas:

Since the VIs are in your user.lib and instr.lib I would keep those in a separate repository. That way you only have to keep up with the reusable code in one place.

You could save your libraries as application distribution (in LV 8 and later this can be done with via a source distribution). Another option is to use VIPM to create a VI Package Configuration (VIPC).

More on the VIPC files: <http://blog.jkisoft.com/products/secrets-of-the-vipc/>

Fab

Like (0)



Lul May 9, 2011 9:50 AM (in response to FabiolaDelaCueva)

Fab,

Thx for your reply.

My problem is not on `_how_` to utilize revision control (SVN does quite a good job here), but how to keep the app revision consistent with the lib revision. So to keep v3 of app A referring to v43 of a given device driver, while v4 refers to v 46 AND while v21 of app B refers to v41 of that driver.

Enlarge this to about 10 different, but similar test sequencer apps, about 8 different device drivers, several other 'private libraries' in user.lib and those apps consisting of some 100 to more than 1000 modules. Not to talk about other related code like databases, MS-word for reporting, several system helper tools like those from the Sysinternals suite etc. pp.

And as I am working in a highly regulated environment, I have to take special care of version control and have to provide traceability info...

Like (0)



ElijahK May 9, 2011 9:58 AM

Just so you know, I've moved the latest version of this content here: <http://zone.ni.com/devzone/cda/tut/p/id/4114>

I'm in the process of changing the scope of this community page to the specifics of integration with third-party tools.

Like (0)



FabiolaDelaCueva May 9, 2011 10:42 AM (in response to Lul)

Lul,

I believe you can use the VI Package Configuration files with SVN to provide the traceability information you need.

Here is a post in LAVAG of someone in a similar situation than yours (LabVIEW 7.1, figuring out how to reuse code in user.lib): <http://lavag.org/topic/13492-reuse-packages-and-scc/> that might help.

Fab

Like (0)



justin.goeres May 9, 2011 10:52 AM (in response to Lul)

Hi Lul,

As Fabiola pointed out, what you describe is pretty much the exact use case that VIPM Professional is designed

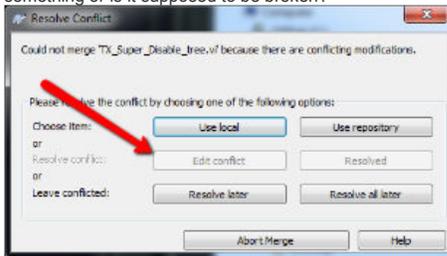


Like (0)



soupy Oct 17, 2011 6:14 PM

This works well when resolving a conflicted file (a file marked with the yellow caution sign glyph), but how do i get it to work when resolving a conflict during an svn merge? The button seems to be grayed out. Am i missing something or is it supposed to be broken?



My solution right now is to use the "resolve later" action and then resolve it later. I would be so much easier to do this in one step...

Like (0)

ElijahK Oct 19, 2011 9:10 AM (in response to soupy)

Did you configure Tortoise for both diff and merge operations? Does this same limitation exist with other file-types, or is it specific to VIs?

Like (0)



JamesMcN Jan 10, 2012 8:51 AM

Simple thing but had me head scratching for a while! Using TortoiseSVN on 64 bit and so you have to use the 64 installer for TSVN I found:

1. Doesn't like the quote marks!
2. Not a simple copy and paste, need to change to Program Files (x86)
- 3.

Hopefully that saves someone a few minutes.

Like (0)



ElijahK Jan 10, 2012 9:17 AM (in response to JamesMcN)

Thanks for posting that James. So is this correct for 64-bit?

C:\Program Files (x86)\National Instruments\Shared\LabVIEW Compare\LVCompare.exe %mine %base -nobdcsm -nobdpos

Like (0)

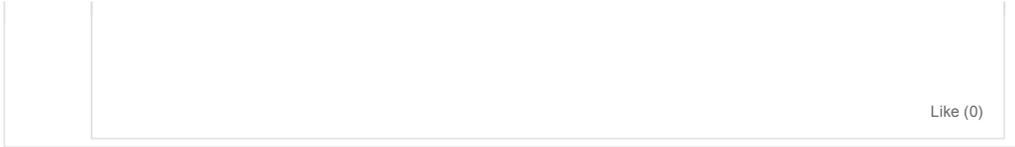
JamesMcN Jan 10, 2012 9:25 AM (in response to ElijahK)

Yes, I can now copy that directly in.

EDIT: Not sure if it is necessary to differentiate at this point that this is with 32 bit LabVIEW on 64 bit OS. Not sure what happens with the compare location on 64 bit LabVIEW.

Like (0)

crelf Apr 15, 2012 12:56 PM (in response to JamesMcN)



Like (0)

